

Appl. No. 10/099,777  
Substitute Appeal Brief dated 08/27/2007  
Reply to Office Action of 07/26/2007

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re: Application of:	:
Brown et al.	:
	: Before the Examiner:
Serial No: 10/099,777	: Cam Linh T Nguyen
	:
Filed: 03/14/2002	: Group Art Unit: 2171
	:
Title: APPARATUS AND METHOD	: Confirmation No.: 4836
OF EXPORTING FILE SYSTEMS	:
WITHOUT FIRST MOUNTING THE	:
FILE SYSTEMS	:

**RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF**  
**UNDER 37 C.F.R. 41.37**

Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

Sir:

This fourth substitute Appeal Brief is being submitted in response to the  
NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF of July 26, 2007.

AUS920010866US1

BRIEF FOR APPLICANTS - APPELLANTS

(i)

Real Party in Interest

The real party in interest is International Business Machines Corporation (IBM), the assignee.

(ii)

Related Appeals and Interferences

There are no other appeals or interferences known to appellants, appellants' representative or assignee, which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(iii)

Status of Claims

Claims 1 - 20 have been finally rejected. This appeal involves all the rejected claims.

(iv)

Status of Amendment

No amendment was filed after the Final Office Action of March 8, 2005.

(v)

Summary of Claimed Subject Matter

The present invention, as claimed in Claim 1, provides a method of exporting file systems. The method comprises the steps of: consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems (see page 14, lines 22 – 24 and page 15, lines 18 – 20 and item 730 of Fig. 7), the mount point being the point at which the file systems are mounted on a computer system (see page 13, lines 9 – 12, lines  
AUS920010866US1

22 and 23, page 14, lines 4 – 16 as well as item 620 of Fig. 6); and exporting the file systems (page 14, line 25 and 26, page 16, lines 17 – 19, lines 24 – 26, items 814 and 818 of Fig. 8).

The present invention, as claimed in Claim 6, provides a computer program product on a computer readable medium for exporting file systems. The computer program product comprises: code means for consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems (see page 14, lines 22 – 24 and page 15, lines 18 – 20 and item 730 of Fig. 7), the mount point being the point at which the file systems are mounted on a computer system (see page 13, lines 9 – 12, lines 22 and 23, page 14, lines 4 – 16 as well as item 620 of Fig. 6); and code means for exporting the file systems (page 14, line 25 and 26, page 16, lines 17 – 19, lines 24 – 26, items 814 and 818 of Fig. 8). The code means of the claim are the steps outlined on page 16, lines 1 – 26 as well as in Fig. 9.

The present invention, as claimed in Claim 11, provides an apparatus for exporting file systems. The apparatus comprises: means for consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems (see page 14, lines 22 – 24 and page 15, lines 18 – 20 and item 730 of Fig. 7), the mount point being the point at which the file systems are mounted on a computer system (see page 13, lines 9 – 12, lines 22 and 23, page 14, lines 4 – 16 as well as item 620 of Fig. 6); and means for exporting the file systems (page 14, line 25 and 26, page 16, lines 17 – 19, lines 24 – 26, items 814 and 818 of Fig. 8). The means of the claim are the steps outlined on page 16, lines 1 – 26 as well as in Fig. 9 when executed by processor 202, processor 204 of Fig. 2 or processor 302 of Fig. 3.

The present invention, as claimed in Claim 16, provides a computer system for exporting file systems. The computer system comprises: at least one storage device (local memory 209, hard disk 232 of Fig. 3, main memory 304, disk 326, tape 328, CD-ROM 330 or memory 324 of Fig. 3) for storing code data; and at least one processor (processor 202, processor 204 of Fig. 2 or processor AUS920010866US1

302 of Fig. 3) for processing the code data to consult a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems (see page 14, lines 22 – 24 and page 15, lines 18 – 20 and item 730 of Fig. 7), the mount point being the point at which the file systems are mounted on the computer system (see page 13, lines 9 – 12, lines 22 and 23, page 14, lines 4 – 16 as well as item 620 of Fig. 6), and to export the file systems (page 14, line 25 and 26, page 16, lines 17 – 19, lines 24 – 26, items 814 and 818 of Fig. 8).

(vi)

Grounds of Rejection to be Reviewed on Appeal

**Whether claims 1 - 20 were properly rejected under 102(a) as being anticipated by Vahalia et al.**

(vii)

Arguments

**Claims 1, 6, 11 and 16**

Vahalia et al. purport to teach a method of: (1) providing NFS clients with read/write access to read from and write into file systems; and (2) performing failure recovery of a failed server.

In the method of providing NFS clients with read/write access to read from and write into file systems, each file system is assigned to a particular server in a network of servers. Any server in the network may receive a file access request from any NFS client. If the file system that is to be accessed is assigned to the server that receives the request, that server will provide the access. But, if the file system that is to be accessed is not assigned to the server that receives the request, the server will forward the request to the server to which the file system is assigned. This scheme obviates the need to provide coherency since only the server to which the file is assigned will allow changes to any file in the file system and will presumably permit only one client to make changes to the files in the file

AUS920010866US1

system at a time. Further, the scheme provides a certain level of load balancing as only a server to which a file system is assigned will process the request and presumably access requests will be sent to different file systems assigned to different servers in the network. To determine to which one of the servers the file system is assigned, a file that contains file system/computer assignment information is consulted.

In the method of performing failure recovery of a failed server, one of the servers monitors the rest of the servers to detect failures. When a failure of a server is detected, the file systems that were assigned to that failed server are re-assigned to an operational server. Thus, requests can always be processed.

The Examiner stated that in col. 13, lines 19 – 26, Vahalia et al. teach the step of consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a computer system. Appellants respectfully disagree.

In col. 13, lines 19 – 39, Vahalia et al. disclose:

Each of the data movers in a cluster has a directory of the file systems in the cluster and a data base of the mount points for the file systems and the data mover owner of each read-write file system. When any data mover in a cluster receives a request for access, it checks the data base, and if it finds that the file system is a read/write file system having a different primary data mover, then the data mover forwards the client request to the primary data mover. Read requests, write requests, mount requests, and lock requests are examples of data access requests that are forwarded to the primary data mover for the file system to be accessed. If a data mover receives a request for access to a file system in the cluster and finds that the request is a read-only request or the data mover finds that it is the primary data mover for the file system, then it accesses the file system directly. Look-up of the file name in the

file directory and management of access authorization for a file system can be distributed between a secondary data mover that receives and interprets a client request for access to the file system and the primary data mover that manages the locks on the file system.

According to the cited passage, therefore, a directory and a database are associated with each server. Each server consults its associated directory and database upon receiving a request for a file in order to determine to which one of the servers the file system containing the requested file is assigned. As mentioned above, if the requested file is in a file system that is assigned to a server other than the request receiving server, the request receiving server will forward the request to the proper server for service; otherwise the request receiving server services the request.

The Examiner further stated that in col. 13, lines 51 – 55, Vahalia et al. teach that the mount point can be attached in the directory structure, so that the tree is traversed from the root to the file to be accessed. Therefore, the Examiner concluded, the file that is located in the directory tree structure corresponds to the file associated to the mount point.

In col. 13, lines 51 – 55, Vahalia et al. disclose:

In step 94, execution branches to step 95 if the file for which access is requested is in a remote file system. This occurs if a "mount point" is reached during file look-up as the file name, which is a path through the directory tree, is traversed from the root of the tree to the file to be accessed.

In the cited passage, the "file" being looked up refers to the requested file and not to the directory and/ or database which, the Examiner earlier stated is consulted. Indeed, the file lookup occurs within the directory. Therefore, whether or not the file that is located in the directory tree structure corresponds to the file associated to the mount point is irrelevant in this case since it is the

directory and/or database that is consulted and therefore it is the directory and/or database that needs to be associated to the mount point as per the claims.

Hence, Vahalia et al. do not teach the step of **consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a computer system** as claimed by the Examiner.

The Examiner further stated that Vahalia et al. teach the step of exporting the file systems in col. 13, lines 55 – 58. Appellants kindly disagree.

In col. 13, lines 55 – 58, Vahalia et al. disclose:

In step 95, the data mover sends the request to an NFS thread for export to the remote file system corresponding to the "mount point."

According to the cited passage therefore, it is the request that is exported and not "the file systems" as in the claimed invention.

Consequently, Vahalia et al. do not teach the step of **exporting the file systems** as in the claimed invention.

It is a well settled law that in considering a Section 102 rejection, all the elements of the claimed invention must be disclosed in a single item of prior art in the form literally defined in the claim. *Jamesbury Corp. v. Litton Indus. Products*, 756 F.2d 1556, 225 USPQ 253 (Fed. Cir. 1985); *Atlas Powder Co. v. Dupont*, 750 F.2d 1569, 224 USPQ 409 (Fed. Cir. 1984); *American Hospital Supply v. Travenol Labs.*, 745 F.2d 1, 223 USPQ 577 (Fed. Cir. 1984).

Since Vahalia et al. do not teach all the elements of the claimed invention in the form literally defined in the claim, Appellants submit that the claimed invention is not anticipated by Vahalia et al.

### **Claims 3, 8, 13 and 18**

Claims 3, 8, 13 and 18 include the limitations "wherein the file systems are exported without first being mounted."

AUS920010866US1

The Examiner stated that “Vahalia teaches that to reduce the loading of the cached disk storage, the data mover have [sic] a local file directory, which can reference to other storage. Therefore, the files are exported without first being mounted.” Appellants disagree.

Appellants note that the Examiner did not point to any passage in the disclosure of Vahalia et al. where it is taught that files are exported without first being mounted. The Examiner did not point to any passage because Vahalia et al. do not teach such step. As mentioned above, the disclosure of Vahalia et al. is directed toward a method of: (1) providing NFS clients with read/write access to read from and write into file systems; and (2) performing failure recovery of a failed server. Therefore, there is no reason for Vahalia et al. to teach the limitations of **wherein the file systems are exported without first being mounted** as in the claimed invention.

Consequently, Appellants submit that the claims are not anticipated by Vahalia et al.

#### **Claims 4, 9, 14 and 19**

Claims 4, 9, 14 and 19 include the limitations “wherein the file is an extended attribute file.”

The Examiner stated that Vahalia disclose in col. 17, lines 15 – 23 the node that reference to other node is a link that can reference to other location. Appellants disagree.

In col. 17, line 15 – 23, Vahalia et al. disclose:

If a file is referenced in a file system but is not local to the file system, then the file authorization and ownership information identifies a remote file system to which can be exported a client request for the file. For example, NFS file names are defined as a path through a directory tree representing the file system including the file, and a reference to a remote file system may be included at



any node on the path. Nodes referencing remote file systems are called "mount points."

In the above-reproduced passage, therefore, Vahalia et al. disclose that a mount point is a node that references a remote file system. However, Vahalia et al. do not teach that **the file is an extended attribute file** as in the claimed invention.

Hence, Claims 4, 9, 14 and 19 are not anticipated by Vahalia et al.

### **Claims 5, 10, 15 and 20**

Claims 5, 10, 15 and 20 include the limitations of "wherein each mount point has an extended attribute file."

In this case, the Examiner pointed to the same passage that was used to reject Claims 4, 9, 14 and 19 to reject Claims 5, 10, 15 and 20. Appellants again disagree.

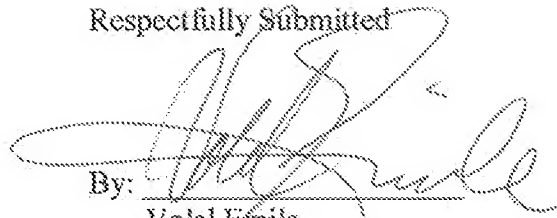
As mentioned above, In the cited passage, Vahalia et al. disclose that a mount point is a node that references a remote file system. But, Vahalia et al. do not teach that **each mount point has an extended attribute file** as in the claimed invention.

Consequently, Appellants submit that Claims 5, 10, 15 and 20 are not anticipated by Vahalia et al. as claimed by the Examiner.

In view of the foregoing, Appellants request reversal of the rejection and passage to issue of the claims in the Application.

Appl. No. 10/099,777  
Substitute Appeal Brief dated 08/27/2007  
Reply to Office Action of 07/26/2007

Respectfully Submitted

By:   
Volel Emile  
Attorney for Applicants  
Registration No. 39,969  
(512) 306-7969

(viii)

Claims Appendix

1. (Previously presented) A method of exporting file systems comprising the steps of:

consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a computer system; and

exporting the file systems.

2. (Original) The method of Claim 1 wherein the needed information is names of devices within which the file systems are located.
3. (Original) The method of Claim 2 wherein the file systems are exported without first being mounted.
4. (Original) The method of Claim 3 wherein the file is an extended attribute file.
5. (Original) The method of Claim 4 wherein each mount point has an extended attribute file.
6. (Previously presented) A computer program product on a computer readable medium for exporting file systems comprising:

code means for consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a computer system; and

code means for exporting the file systems.

7. (Original) The computer program product of Claim 6 wherein the needed information is names of devices within which the file systems are located.
8. (Original) The computer program product of Claim 7 wherein the file systems are exported without first being mounted.
9. (Original) The computer program product of Claim 8 wherein the file is an extended attribute file.
10. (Original) The computer program product of Claim 9 wherein each mount point has an extended attribute file.
11. (Previously presented) An apparatus for exporting file systems comprising:  
  
means for consulting a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a computer system; and  
  
means for exporting the file systems.
12. (Original) The apparatus of Claim 11 wherein the needed information is names of devices within which the file systems are located.

13. (Original) The apparatus of Claim 12 wherein the file systems are exported without first being mounted.
14. (Original) The apparatus of Claim 13 wherein the file is an extended attribute file.
15. (Original) The apparatus of Claim 14 wherein each mount point has an extended attribute file.
16. (Previously presented) A computer system for exporting file systems comprising:  
  
at least one storage device for storing code data; and  
  
at least one processor for processing the code data to consult a file associated with a mount point of a mounted file system to retrieve needed information to export the file systems, the mount point being the point at which the file systems are mounted on a the computer system, and to export the file systems.
17. (Original) The computer system of Claim 16 wherein the needed information is names of devices within which the file systems are located.
18. (Original) The computer system of Claim 17 wherein the file systems are exported without first being mounted.
19. (Original) The computer system of Claim 16 wherein the file is an extended attribute file.

20. (Original) The computer system of Claim 19 wherein each mount point has an extended attribute file.

(ix)

Evidence Appendix

None.

Appl. No. 10/099,777  
Substitute Appeal Brief dated 08/27/2007  
Reply to Office Action of 07/26/2007

(x)

Related Proceedings Appendix

None.